08

Referencias

Transcripción

[00:00] Le damos cliente.set. No. Es cuenta.setTitular. Y adivinen a quién le vamos a setear. A nuestro objeto tipo titular que es el cliente. Perfecto. De esta forma, si nosotros queremos imprimir el nombre del titular, vamos a darle sysout, vamos a darle un cliente.getNombre. Nosotros vamos a ver que él ya asignó el nombre Diego al nombre de este cliente.

[00:42] Y ahora vamos a ver algo bien curioso. Vamos a ver un sout y le vamos a decir ahora a la cuenta y obtenemos el titular de esa cuenta. No tenemos aquí el método get, por lo tanto simplemente lo generamos. GetTitular. Listo, no hay más ciencia, solamente es un autocompletar, y acá "Ctrl + espacio", getTitular. Y aquí le podemos dar un punto y accedemos a los métodos del objeto cliente que tenemos ahí, en este caso nombre.

[01:27] Y tenemos algo así. Para explicar mejor esto, vamos a ver la presentación. Como vimos anteriormente, está nuestro objeto cuenta, que es la cuentaDeDiego en ese caso, y está el objeto cliente, son dos objetos diferentes. Y ya sabemos que estos objetos son referencia en el caso de cuenta con cliente. En la variable titular, él referencia directamente al objeto cliente. No es que existe un cliente dentro del campo titular.

[02:01] Acuérdense, en Java todo es referencias, por lo cual si yo tengo un método que accede a este referencia, yo voy a tener acceso a todos los atributos de mi objeto cliente. ¿Cómo demostramos eso? De esta manera. Yo vine aquí y le di cuenta.getTitular, y yo ya tengo acceso a todos los métodos y atributos

públicos que tenga esa clase. ¿Quieren comprobarlo? Vamos a guardar y vamos a ejecutar.

[02:37] Ahora encontramos otros errores por el momento. Aumentamos y vemos claramente que nos imprime el mismo valor. Incluso si nosotros solamente imprimimos cliente, vamos a ver aquí, le quitamos esto de acá, vamos a imprimirle directamente el objeto, el objeto cliente y el objeto getTitular. Guardamos, ejecutamos y vemos que es el mismo objeto al cual está haciendo referencia.

[03:16] Incluso hay otra forma también de definirle el cliente. ¿Cuál podría ser? Yo podría decirle que mi cliente, vamos a ponerle cliente 2, va a ser igual a cuenta.getTitular. ¿Qué les parece eso? Y acá le vamos a poner titular. Y aquí voy a imprimir este titular. Imprimo el titular. ¿Será el mismo objeto? ¿Ustedes creen que sea el mismo objeto? Vamos a guardar. Imprimimos y tenemos aquí el mismo objeto en los tres casos.

[04:12] ¿Por qué? Porque todos referencian a un único objeto en la memoria de Java que es este de acá, este new cliente que yo di aquí. El único new cliente que yo di fue este de aquí, por lo tanto todo esto son solo referencias a este objeto new cliente. Es de esta forma que ya vemos un poco más de la utilidad de mantener nuestro alcance cerrado aquí con respecto a nuestros modificadores de acceso.

[04:48] Ya sabemos que declarando una variable privada, ella no es accesible desde cualquier otra parte del código. Incluso aquí nos olvidamos, nosotros podemos hacerle tranquilamente aquí un private también. Ahora el cliente ya no se puede asignar directamente desde otra parte fuera del código sino que tiene que ser exclusivamente a través del método set del titular.

[05:14] ¿Qué es lo que queda pendiente aquí? Simplemente reemplazar nuestras variables que hemos estado accediendo directamente por los respectivos métodos, que en este caso sería un getSaldo. Aquí yo estoy

asignando el valor, puede ser un setSaldo, pero sabemos que no tengo un setSaldo. ¿Cuál es el método para asignarle saldo a esa cuenta? Es el método depositar.

[05:42] Ya vemos ahí la importancia de no haber creado un método getSaldo. Yo no quiero manipular directamente mi saldo. Mi saldo es manipulable a través de los métodos depositar y retirar. Pueden practicar ahí también en su cuenta, modificando manualmente cada valor y conforme ustedes vayan viendo ahí la necesidad de implementar algún nuevo set o get para algún campo, lo implementan.

[06:08] Recuerden: no es necesario generar para todos los campos aquí, solamente si ustedes de verdad lo necesitan.